**Problem analysis**

**Problem specification**

|  |  |
| --- | --- |
| **Client:** | * The coordinator of the Department of Transformation and Improvement from Universidad Icesi. |
| **Users:** | * Employees from the department of Transformation and Improvement in Universidad Icesi. * Representants of every other department inside Universidad Icesi. |
| **Functional requirements:** | * Req#1: Register a collaborator from Universidad Icesi * Req#2: Register a department from Universidad Icesi. * Req#3: Register a project request. * Req#4: Change the status of a request. * Req#5: Create a Knowledge Management project. * Req#6: Create a Transformation/Improvement project. * Req#7: Close a project. * Req#8: Display the information of the last 5 projects (based on query date) of each team member in matrix form. * Req#9: Access the detailed information of a project. * Req#10: Review the efficiency of a collaborator from the Department of Transformation and Improvement * Req#11: Review the efficiency of a determined project. * Req#12: Review the efficiency of a request. * Req#13: Consult the number of projects by type and priority. * Req#14: Consult the number of projects led by every collaborator in the Department of Transformation and Improvement. * Req#15: Consult the number of received requests and managed in a given month. * Req#16: Generate test objects. |
| **Context of the problem:** | * Universidad Icesi has an appointed Department of Transformation and Improvement which focuses on receiving requests for institutional projects or creating new projects, advising, managing all the appropriate information so the university can improve in an orderly and standardized fashion. Currently the Department faces a problem regarding its organization and management of requests and projects, so it looks for a program that relieves this stress. * The coordinator of the Department of Transformation and Improvement wants his employees (and the rest of collaborators in the university) to utilize the program to manage all the information regarding requests, projects, and statistics. A collaborator should be able do register a request and a worker from the Department should be able to change the status request, make a project, close the project, and some other functionalities. The program should also allow the users to check efficiency indicators regarding requests, projects or collaborators enrolled inside the Department of Transformation and Improvement. * The program should work with a simple menu that multiple users can easily understand and utilize properly. It must be able to handle a big amount of information but maintain a high level of organization and coherence. |
| **Nonfunctional requirements:** | * Interoperability: The program should be able to handle multiple users at a time. * High usability: The program should use a menu that makes it easy to understand while entering, editing, or consulting information. * High availability: The program should be working 24/7 and outage risks mitigated. |

**Requirement specifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#1: Register a collaborator from Universidad Icesi. | | |
| **Summary:** | The system will allow the user to register a new collaborator by taking the mandatory (ID, full name, etc.) and optional information (extension number) to create the appropriate instance of a collaborator. To create the collaborator, it must be specified if it is either a general collaborator or a specific one from the Department of Transformation and Improvement. Afterwards, the system will check if the entered collaborator was already registered before and display a “duplicate” message in case it happens. If any other error were to occur, the system will display an invalid information, incomplete information or a general error message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Collaborator Type | int | * The user must select one of the following options:   1. General Collaborator   2. Transformation and Improvement Collaborator * Mandatory |
| ID | String | * Mandatory |
|  | Full Name | String | * Mandatory |
|  | Institutional email | String | * Mandatory |
|  | Extension number | String | * Optional |
| **Results or postcondition** | A new collaborator is registered with all the mandatory information (adding the extension number if it is filled) and a success message is printed. If a collaborator with the same ID was already registered, a “no duplicate” message will be printed. If there any other error occurs within the process because of incomplete information, invalid information or any other general error, the appropriate error message will be displayed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Successful registration | String | A message stating that the collaborator was successfully registered, stating which type was just registered. |
| Duplicate collaborator | String | A message displaying that a collaborator with the entered ID has already been registered. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#2: Register a department from Universidad Icesi | | |
| **Summary:** | The system facilitates the registration of a new department by allowing users to input essential details such as the department's name, address, internal code, and select a responsible collaborator from a menu displaying registered collaborators. In case there are no preregistered collaborators to select from the list an “at least one collaborator” message will be printed. Upon submission, the program checks to ensure the department being registered doesn't already exist in the system. If a duplicate entry is detected, a clear "duplicate" message is promptly displayed, alerting the user to the existing registration. If any provided information is deemed invalid or incorrect, the user is notified with an appropriate error message. Additionally, should any unforeseen errors occur during the registration process a general error message is presented to the user. Finally, upon successful registration without encountering any errors, the system confirms the successful addition of the new department. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Address | String | * Mandatory |
|  | Responsible Collaborator | int | * The user must enter one of the numbers that is displayed representing one of the names of the preregistered collaborators. * Mandatory |
|  | Department’s name | String | * Mandatory |
|  | Internal Code | String | * Mandatory |
| **Results or postcondition** | The system allows you to register a new department using the name, address, internal code, and a responsible collaborator. The responsible collaborator should be selected from a menu displaying the names of the registered collaborators. If there are no collaborators preregistered an “at least one collaborator” message will be printed. Afterwards, the program will check if the Department has already been registered before, displaying a “duplicate” message in case it had been done. If any other additional errors occur an invalid, incorrect information or general error message will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Successful Registration | String | A message stating that the department was registered successfully will be printed. |
| At least one collaborator | String | A message stating that to register a department there must be at least one collaborator preregistered. |
| Duplicate Department | String | A message stating that the department with the entered internal code had already been registered will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#3: Register a project request | | |
| **Summary:** | The program enables a collaborator to initiate a new request by providing the subject, description, and selecting a collaborator from a menu. If no collaborator is assigned to a department, it displays a 'no eligible collaborator' message. It then checks for any existing requests with the same subject in the collaborator's projects to prevent duplicates, showing a message if found. Upon successful validation, it assigns a unique ID, registration date matching the time of the request and automatically makes its status “pending”. If there are any errors or missing information, it prompts a message detailing the issue. In case of any other errors, it shows a general error message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Subject | String | * Mandatory |
|  | Description | String | * Mandatory |
|  | Collaborator’s Name | Int | * The user must enter one of the numbers that is displayed representing one of the names of the preregistered collaborators. * Mandatory |
| **Results or postcondition** | The program will register a new request done by a collaborator taking the subject, description, and the collaborator (selected from a small menu displaying the names of the preregistered collaborators with an assigned department). If there is not at least one collaborator with an assigned department a “no eligible collaborator” message will be shown. Afterwards, the subject will be used to check if there is any other request with the same topic inside the projects done by that collaborator. If both filters are passed, the new request will be assigned a new unique ID, a date of registration (which will be the one the system provides at the time of registration) and it sets its status to “pending”. If there any other errors that have to do with invalid or incomplete information or any other general error, the system will display an appropriate message. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Successful registration | String | A message stating that the request was registered successfully will be printed along with the unique ID. |
| No eligible collaborator | String | A message stating that to send a request there should be at least one collaborator that is already assigned to one department must already exist will be shown. |
| Duplicate request | String | A message showing that that a request with that subject had already been registered previously by that collaborator will pop up. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#4: Change the status of a request. | | |
| **Summary:** | The system presents a list of all registered projects, showing their IDs and subjects so the user selects one from the list. If no requests are preregistered, it displays an “at least one request” message. Once a request is selected, a menu with three status change options appears (can only pick one option). After the status change, a message confirms the successful update and saves the date of assignation. If there's an error due to invalid or incomplete info, or any other issue, the system shows the appropriate message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Request ID | int | * The user must enter one of the numbers that is displayed representing one of the ID and the subject of a request. * Mandatory |
|  | Status change | int | * A number from the following options must be selected:   1. Accepted   2. Rejected   3. Postponed * Mandatory |
| **Results or postcondition** | The system will display the ID and the subject of all the registered projects and the user will have to select one of the displayed options (Accepted, rejected, or postponed). If there were no preregistered request an “at least one request” message will be shown. If there is at least one request, then, a menu showing the three possible options for status change will appear so that the user can select one of the three. Finally, a message displaying that the status of a request has been changed successfully and it will save the date of assignation. If an error related to invalid or incomplete information or any other general error the appropriate message will appear depending on the case. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Successfully changed the status | String | A message stating that the status from the request was changed successfully will be printed. |
| At least one request | String | A message saying that there should be at least one existing request will be shown. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#5: Create a Knowledge Management project. | | |
| **Summary:** | After a request has been accepted, then the system enables the user to input a new knowledge management project along with mandatory details, including the name of the associated accepted request, priority level, project leader, impacted community, and project type. Menus are presented for selecting an accepted request, project leader (from preexisting collaborators), impacted community, and project type. If there's no accepted request or registered collaborator from the DTI, a message directs the user to enter the required information. In case of a duplicate project name, a "duplicate" message is displayed. Upon successful validation, the system generates a unique project ID, a registration date, and an estimated closure date. Any errors related to incomplete or invalid information prompt appropriate error messages. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Accepted request | int | * The user must enter one of the numbers that is displayed representing one of the ID and the subject of an accepted request. * Mandatory |
|  | Priority | int | * The user must select one of the following numbers representing a priority level:   1. High   2. Medium   3. Low   4. Urgent * Mandatory |
|  | Project leader | int | * The user must enter one of the numbers that is displayed representing the ID and full name of an employee from the Department of Transformation and Improvement. * Mandatory |
|  | Project Name | String | * Mandatory |
|  | Impacted community | String | * One of the following options must be selected:   1. Students   2. Professors   3. Administrative * Mandatory |
|  | Type | int | * One of the following options must be selected:   1. Standardization   2. Documentation   3. Optimization * Mandatory |
| **Results or postcondition** | After a request has been accepted, the system will allow the user to create a new knowledge management project using all the mandatory information including the name of the associated accepted request, a priority level, project leader, an impacted community, and a type of project. A menu will be printed to select an accepted request, a project leader (from the preexisting collaborators from the DTI), the impacted community and the type of project. If there is not at least one accepted request or registered collaborator from the DTI a message will be printed telling the user to enter one of each. If the entered name for the project already exists a “duplicate” message will be printed. If all the filters are passed, the system will create a unique ID for the project, a registration date (the same day the registration happened), and an estimated date of closure (Urgent = 5 days, high = 10 days, medium = 30 days, low = 60 days). For other general errors, incomplete information or invalid information, the appropriate error messages will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Created knowledge management project | String | A message stating that the knowledge management project was registered successfully will be printed. |
| At least one accepted request | String | A message stating that there must be at least one accepted request will be printed. |
| At least one collaborator from the Department | String | A message stating that there must be at least one collaborator from the Department of Transformation and Improvement. |
| Duplicate project | String | A message stating that a project with that name already exists will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#6: Create a Transformation/Improvement project | | |
| **Summary:** | After a request has been accepted, the system lets users input a new Transformation/Improvement project, needing details like the associated accepted request name, priority level, project leader, and process code. Menus help pick an accepted request and project leader. If there's none, it asks to enter them. If the project name already exists, it warns about duplicates. After validation, the system creates a unique project ID, registration date (same as entry), and estimated closure date (Urgent = 5 days, High = 10 days, Medium = 30 days, Low = 60 days). Errors or missing info trigger appropriate messages. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Accepted request | int | * The user must enter one of the numbers that is displayed representing one of the ID and the subject of an accepted request. * Mandatory |
|  | Priority | int | * The user must select one of the following numbers representing a priority level:   1. High   2. Medium   3. Low   4. Urgent * Mandatory |
|  | Project leader | int | * The user must enter one of the numbers that is displayed representing the ID and full name of an employee from the Department of Transformation and Improvement. * Mandatory |
|  | Project Name | String | * Mandatory |
|  | Process Code | String | * Mandatory |
| **Results or postcondition** | After the system accepts a request, it will allow the user to enter a new Transformation/Improvement project using all the mandatory information including the name of the associated accepted request, a priority level, project leader, and the process code. A menu will be printed to select an accepted request and a project leader (from the preexisting collaborators from the DTI). If there is not at least one accepted request or registered collaborator from the DTI a message will be printed telling the user to enter one of each. If the entered name for the project already exists a “duplicate” message will be printed. If all the filters are passed, the system will create a unique ID for the project, a registration date (the same day the registration happened), and an estimated date of closure (Urgent = 5 days, high = 10 days, medium = 30 days, low = 60 days). For other general errors, incomplete information or invalid information, the appropriate error messages will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Created Transformation/Improvement project | String | A message stating that the Transformation/Improvement project was registered successfully will be printed. |
| At least one accepted request | String | A message stating that there must be at least one accepted request will be printed. |
| At least one collaborator from the Department | String | A message stating that there must be at least one collaborator from the Department of Transformation and Improvement. |
| Duplicate project | String | A message stating that a project with that name already exists will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#7: Close a project | | |
| **Summary:** | The system allows the user to close a project by selecting it from a menu that displays all the registered projects. If there are no registered projects, the system will prompt that there must be at least one project in order to close one. After the project is selected, the user must enter the date on which the project was or will be closed; this date must be a later date compared to the date of registration or a “earlier closing date” message will be printed. If there is any error in the format or the information is incomplete an invalid or incomplete information message will appear. Any other error will be shown as a general error. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Project | int | * The user must enter one of the numbers that is displayed representing one of the ID and the name of a project. * Mandatory |
|  | Closing Date | Calendar | * Must be a date later than the registration date. * Mandatory |
| **Results or postcondition** | The system allows the user to close a project by selecting it from a menu that displays all the registered projects. If there are no registered projects, the system will prompt that there must be at least one project in order to close one. After the project is selected, the user must enter the date on which the project was or will be closed; this date must be a later date compared to the date of registration or a “earlier closing date” message will be printed. If there is any error in the format or the information is incomplete an invalid or incomplete information message will appear. Any other error will be shown as a general error. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Closed project | String | A message stating that the project was closed successfully will be printed. |
| At least one project | String | The system will print a message stating that there is not one project to close. |
| Earlier closing date | String | A massage stating that the closing date entered is an earlier date than the registration date will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#8: Display the information of the last 5 projects (based on query date) of each team member in matrix form. | | |
| **Summary:** | The system provides users with an option to display all necessary information for the last 5 projects allocated to each team member in matrix form via a menu. If the entered query date lacks at least one project for one of the collaborators in DIT with a posterior date, an "empty date" message appears. The matrix includes collaborator names, project IDs, priorities, and estimated deadlines, with project lists per collaborator ordered not solely by priority. If no preregistered projects exist before selecting this option, an "at least one project" message is displayed. Any other errors prompt a general error message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Consultation date | Calendar | * The users are required to input the date that will determine the projects to be displayed. * Mandatory |
| **Results or postcondition** | The user will be able to select from the menu the option to display all the needed information of the last 5 projects (based on query date) allocated to each team member in matrix form. If the entered query date doesn’t have at least 1 project for one collaborator in the DIT with posterior date, a “empty date” message will be printed. The included information in the matrix must be the collaborator, the ID of the project, the priority and the estimated deadline. Additionally, the project list per collaborator must be ordered not only by priority. If there are no preregistered projects before selecting this option an “at least one project” message will be printed. If there is any other type of error, a general error message will be shown. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the community was registered successfully will be printed. |
| At least one project | String | A message stating that there should be at least on registered project to show the project information in matrix form. |
| Empty Date | String | A message stating that the entered query date doesn’t show any projects for any collaborator in the DIT with a registered project with posterior date will be shown. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#9: Access the detailed information of a project. | | |
| **Summary:** | After selecting the option to display project information based on a query date, the user can choose one of the IDs printed in the matrix to view detailed project information. If the user inputs "0", this step is skipped. Before selecting the ID, the system verifies there was at least one preregistered project and one shown project based on the query date (using the criteria of "at least one project" and "Empty Date" stated in Req#8). Subsequently, it searches for the project using its ID; if not found, a "nonexistent project" message is printed. Any errors like invalid or incomplete information prompt appropriate messages. If there are any other errors, a general error message is displayed. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Project ID | String | * The user should enter the ID of a project that was shown from the project visualization matrix from REQ#8 * Mandatory |
| **Results or postcondition** | After the user has selected the option in Req#8 (display the information of projects in a matrix based on a query date) then the user must be able to select one of the ID’s printed in the matrix to print all the information regarding that project. If the user enters a “0”, this step is skipped. Before selecting the ID, the system will verify there was at least one preregistered project and one shown project based on the query date (using the criteria of “at least one project” and “Empty Date” stated in Req#8). Finally, it will look for the project using its ID (if not found, a “nonexistent project message is printed). If there are any other errors like invalid or incomplete information, the appropriate message will be shown. On the other hand, if there are any other errors, a general error message will pop up instead. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Project information | String | A message containing all the information from the entered project will be printed. |
| Nonexistent project | String | A message stating that there the entered ID doesn’t match with the ID of any of the preregistered projects will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#10: Review the efficiency of a collaborator from the Department of Transformation and Improvement | | |
| **Summary:** | The system displays a menu with names of collaborators who have had at least one project assigned to them. Upon selecting a collaborator, the system calculates and prints their efficiency using the formula stated in the postconditions box. If no projects are registered, it prompts an "at least one project" message. Similarly, if there are no DIT collaborators who haven't been assigned as leaders to any project, it prints an "at least one leader" message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Project leader | int | * The user must select an int that represents a collaborator from the DIT that was assigned at least one project. * Mandatory |
| **Results or postcondition** | The system will print the efficiency of a collaborator from the DIT selected from the menu that displays names of the collaborators that had at least one project assigned to them. The efficiency of the collaborator is calculated using the following formula:  If there were no registered projects, an “at least one project” message will be printed. Likewise, if there were no DIT collaborators that weren’t assigned as leaders to any project, then an “at least one leader” message will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Collaborator Efficiency | String | A message showing the collaborator efficiency of the selected DIT collaborator will be printed. |
| At least one project | String | A message stating that there should be at least on registered project will be printed. |
| At least one leader | String | A message showing that there must’ve been at least one collaborator as assigned leader will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#11: Review the efficiency of a determined project | | |
| **Summary:** | The program lets the user evaluate the efficiency of a Transformation/Improvement project by selecting one from the displayed list of closed projects. It calculates efficiency using the formula shown in the postconditions. If there are no registered projects, it prints an “at least one project” message. Additionally, if there are no closed projects, it prints an “at least one closed project” message. For any other errors like incomplete or invalid information, or general errors, the program displays the appropriate message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Project | int | * The user must select one of the preregistered projects that have already been closed which will be displayed in an orderly menu. * Mandatory |
| **Results or postcondition** | The program will allow the user to review the efficiency of a Transformation/Improvement project by selecting one from the displayed list of closed projects and using the following formula:    If there is not at least registered project, an “at least one project” message will be printed. Furthermore, if there is not at least one closed project, a “at least one closed project” message will be printed. If there are any other errors like incomplete information, invalid information or any other general error, the appropriate message will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Project Efficiency | String | A message showing the project efficiency of the selected project will be printed. |
| At least one project | String | A message stating that there should be at least on registered project will be printed. |
| At least one closed project | String | A message stating that there should be at least on registered closed project will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#12: Review the efficiency of a request | | |
| **Summary:** | The system enables users to check the efficiency of a request by selecting one of the requests that has already been accepted or rejected. The efficiency is calculated using the formula stated in the postconditions. If at least one request is registered and has had its status changed. If these conditions are not met, the appropriate message is displayed. If the entered information is incomplete, invalid, or any other general error occurs, the system prints the appropriate message. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Request | int | * The user must select an option from the displayed list of requests which had their status changed to either rejected or accepted. * Mandatory |
| **Results or postcondition** | The system will allow the user to consult the efficiency of a request if there was at least one registered request and a request which status has been changed already. If neither of those conditions are met, the appropriate message will be printed. If the filters are passed, the user will be able to enter one of the displayed requests which had already been accepted or rejected to review the efficiency. The efficiency of a request is calculated using the following formular:  If the entered information was incomplete, invalid or any other general error happened in the process, the right message will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Request Efficiency | String | A message showing the efficiency of the selected request will be printed. |
| At least one request | String | A message stating that there must be at least one preregistered request will be printed. |
| At least one accepted or rejected request | String | A message stating that there must be at least one accepted or rejected request will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#13: Consult the number of projects by type and priority | | |
| **Summary:** | The system displays the number of registered Transformation/Improvement and Knowledge Management projects, categorized by project type and priority level. If there are no registered projects, it prints a “no registered projects” message. For any other general errors, an appropriate message is displayed. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| N/A | N/A | * N/A |
| **Results or postcondition** | The system will show the number of Transformation/Improvement and Knowledge Management projects that are registered at the moment. The information will be divided by the type of project and the four types of priority. If there is not at least one registered project, an “no registered projects” message will be printed. If any other general error were to happen, a message will be shown in that case. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Number of projects | String | A message showing the number of projects by type and priority levels will be printed. |
| No registered projects | String | A message stating that there is not one registered project will be shown instead of counting. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#14: Consult the number of projects led by every collaborator in the Department of Transformation and Improvement | | |
| **Summary:** | The system lets users check the number of projects led by each DTI collaborator, showing their full name and ID alongside a counter. If there are no registered DTI collaborators, it displays a message saying so. If there are DTI collaborators but none are project leaders, it prints a corresponding message. For other errors, a general error message is shown. | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| N/A | N/A | * N/A |
| **Results or postcondition** | The system will allow the user to consult the number of projects led by every DTI collaborator. The message will contain the full name and ID of every collaborator with the counter by its side. If there is not at least one registered DTI collaborator, a “no registered DTI collaborators message will be shown”. If there are registered DTI collaborators, but none of them are a project leader a “no registered project leaders” will be printed. If there are any other errors a general error message will be shown. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Number of led projects per collaborator | String | A message showing the number of projects led by every DTI team member will be printed. |
| No registered DTI collaborators | String | The system will show a message saying that there are no registered DTI collaborators. |
| No registered project leader | String | A message showing that there must’ve been at least one collaborator as assigned leader will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#15: Consult the number of received requests and managed in a given month | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| N/A | N/A | * N/A |
| **Results or postcondition** |  | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the community was registered successfully will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#16: Generate test objects | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| N/A | N/A | * N/A |
| **Results or postcondition** |  | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the community was registered successfully will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |